Applied Data Science Master's degree thesis

Measuring the Similarity Between Graph Representations of GIS Workflows in Geo-analytical QA

Student:
Anqi Jiang

Supervisor:
Simon Scheider

Department of Information and Computing Sciences

Utrecht University,

Netherlands,

July 29, 2023

Contents

1 Introduction

In the field of Geographic Information Systems (GIS), workflows are commonly used to define and automate geo-analytical tasks. For example, to assess regional flash flood risk using maps and hydrological models, [1] propose an unified workflow using the ArcGIS Model(https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview). The systematic and consistent analysis of different watersheds can be ensured by using workflows.

A Question Answering (QA) system is a type of information retrieval system designed to provide direct answers to user queries[2]. The integration of QA systems into the field of GIS brings innovative perspectives to this domain[3]. The concept of workflow is also applied to geo-analytical question answering, which is a problem about how to represent the analytical potential of a data set to answer a question, based on good theories about spatial problems and the possibility of operational transformations offered by GIS[3]. When it comes to geo-analytical question answering, it becomes necessary to map questions to workflow graphs with the help of semantic concepts. The generated workflow graph provide a visual representation of the steps involved in answering a geo-analytical query.

QuAnGIS is an example of a geo-analytical QA system that generates workflows based on questions. Each node in the workflow graph represents a specific geodata set or geo-analytical tool, while the edges represent the flow of data between these operations. To evaluate the system's performance and conduct retrieval experiments, it is crucial to assess the validity and quality of the generated workflows. This assessment involves comparing the generated workflows with expert workflows, typically through a recall test.

For instance, suppose a user asks a question like, "How much is Tom exposed to green while running through Amsterdam?". The suggested workflow is shown in Figure 1 [3]. This workflow graph provides a comprehensive step-by-step guide on how to answer the given geo-analytical query using input and output geographic data interpreted by the core concepts[4] and ArcGIS operator names (https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview). Before delving into the specifics, it's important to understand the Core Concept Data Ontology, which encompasses three dimensions: core concepts, geometric layer types, and measurement levels[5]. In the context

of core concepts, we often deal with objects, fields, networks, and events. Geometric layer types include raster, vector, and others, and they can be measured at different levels[4]. For instance, in the workflow example, the field raster represents a focal density measurement - a measurement of tree density around a moving window for each location in space, and the trees themselves are discrete objects represented as object vectors. However, the terminology 'green' in the question can lead to varied interpretations in terms of geospatial concepts, and in turn, different workflow solutions. For example, it can refer to a map containing parks or a collection of trees. Thus, the similarity approach is a crucial tool for evaluating these diverse workflow solutions.

Figure 1: Suggested Workflow

Note. From "Geo-analytical question-answering with GIS," by S. Scheider, E. Nyamsuren, H. Kruiger, and H. Xu, 2021, International Journal of Digital Earth, 14(1), p.11. (https://doi.org/10.1080/17538947.2020.1738568). Copyright 2021 by Taylor & Francis.

It is possible that the system produces workflows that may not match the expert workflows exactly but are similar or close in nature. Hence, it becomes important to have a workflow similarity measure that can distinguish similarity relevant workflows from irrelevant workflows. The objective of this thesis is to explore and experiment with various graph similarity measures specifically tailored for this purpose.

One such promising approach can be the adoption of vector embedding methods, such as the word2vec model. This method for learning vector representations of words has witnessed significant success in the realms of Question-Answering (QA) and Natural Language Processing (NLP)[6]. Given their proven efficacy in measuring similarity, these methods warrant exploration for their potential application to workflows. In this context, it is essential to underscore that such an approach has rarely been employed specifically for workflow similarity, thereby highlighting the novelty of this

approach.

In light of the aforementioned discussion, the thesis delineates the central research subquestions.
1) How might we utilize graph embedding methods to assess the similarity among GIS workflows?
2) How does this approach compare with conventional methods, such as graph serialization, in the context of assessing workflow similarity?

2 Literature Review

The effective utilization of Geographic Information System (GIS) workflows relies on the integration of two key components: GIS tools and data types. To handle the similarity between GIS workflows represented as graphs, graph embedding techniques have emerged as a promising approach. The graph serialization method is introduced as a means of comparison with the graph embedding approach.

2.1 Conceptual Models of GIS

The identification of valid nodes within the GIS workflow heavily relies on the geo-analytical tools and data types present in the workflow. These elements serve as the focal points for constructing the wrokflows accurately. Core concepts are formally incorporated into the Core Concept Data Type (CCD) ontology, which serves as a semantic data type system dedicated to capturing the various ways in which core concepts are represented in geodata, taking into account their different levels of measurement[5] .Each data type set contains three dimensions, including core concepts, geometric layer types, and measurement levels[5].

Through understanding the differences between a Field Raster and a Object Vector, we can shed light on various CCD types. Distinguishing between geographic layer types is based on the most prominent geographic data type raster or vector as a layer is the basic concept of every GIS to compare and combine information based on spatial coincidence[5]. Raster data is a type of geospatial data that is stored in a grid raster format and is ideal for continuous data, while vector data is ideal for representing discrete features

of the world, like buildings. Fields and objects are two properties of the core concepts. The former, whose domains are locations that allow distance evaluation, do not change position but change in time, while the latter can change their position and quality at each moment of time while maintaining their identity[5].

2.2 Semantic Workflows Representation

Semantic workflow representations enrich the workflow formats by adding metadata and constraints to the individual elements of the workflow, and several semantic web languages have been proposed to capture relationships and dependencies between different data elements[7].

One of the semantic web language is Resource Description Framework(RDF), which is a W3C standard and is used to describe workflow semantics[8]. The expert workflows and the automatically generated workflows used in this thesis are all represented in RDF format.

2.3 Graph Similarity Measures

A promising approach for measuring similarities between graphs is to employ graph embedding models within the field of machine learning. One such model is graph2vec, which generates vector representations for entire graphs. By capturing the structural information of graphs in vector space, graph2vec enables comparisons between graphs based on the presence of common subgraphs.

Several studies have explored the effectiveness of graph embedding techniques, including graph2vec, in various domains. For instance, [9]propose and develop a neural embedding framework named graph2vec inspired by the success of neural document embedding models. Before introducing the document embedding method, it is necessary to understand the background of word embedding. The word2vec model leverages a straightforward and efficient feed-forward neural network architecture, known as "skip-gram", to learn distributed representations of words[6]. The skip-gram model employed capitalizes on the concept of context, where in this model, 'context' refers to a fixed number of words that surround a given target word[9].

Neural document embedding models like doc2vec model excel at capturing the composition of words or word sequences within documents to generate embeddings. The doc2vec algorithm comes in two versions: Paragraph Vector-Distributed Memory (PV-DM) and Paragraph Vector-Distributed Bag of Words (PV-DBOW). The latter is an instantiation of the skip-gram model, whereas the former is not considered a variant of the skip-gram model. Consequently, PV-DM does not share a direct correlation with the graph2vec technique, we only need to consider the PV-DBOW version implementing the skip-gram model[9].Building upon this concept, graph2vec considers an entire graph as a document and treats the rooted subgraphs as words that collectively form the document.

Another approach for measuring similarities between graphs is the graph serialization method. An approach for measuring XML similarity involves serializing XML data into XML node sequences using a tree-traversal order, and subsequently employing the edit distance method to compare structural information, serves as an inspiration for this study[10]. This suggests the potential to serialize the structure of a workflow graph into a string following a specific sequence, after which the edit distance method can be leveraged to ascertain the similarity between two such sequences.

3 Research Data

The research data comprises expert workflows as well as automatically generated workflows. These data sources were chosen for their relevance and applicability to the study of workflow similarity assessment.The expert workflows provide tutorial examples of how GIS tasks are performed, while the automatically generated workflows allow for the exploration of a wide range of potential solutions to the same tasks.

A dataset comprising 14 expert GIS workflows were reproduced from online GIS tutorials. The source data for both the expert workflows and the generated workflows utilized in this study can be accessed at the following website:[11]

Moreover, a total of 168 workflows were generated based on the tutorials to ensure consistency with the expert workflows. The generated workflows,

utilizing the loose programming of GIS, is significantly influenced by the quality of the semantic model employed to describe GIS functionality[4]. Such semantic models are fundamentally grounded in the types of CCD ontology, which formulates the input and output constraints of a tool - also known as its operational signature. These CCD ontology types, providing the semantic foundation, play an integral role in workflow construction research[4]. For this purpose, a tool repository was abstracted from annotated workflows, and this repository was subsequently utilized to construct new generated workflows using APE(Automated Pipeline Explorer)[12]. Furthermore, the abstracted tool repository comprises both super tools, which are inherently more complex, and simple tools, each equipped with a CCD signature.

4 Data Preparation

Prior to calculating the similarity between the expert workflows and the generated workflows using various methods, it is necessary to perform specific data processing steps. These steps are crucial to ensure that the data is in a suitable format for the similarity calculations, and to ensure that the results are meaningful and can effectively answer the research questions.

4.1 Retrieve Graphical Representations

To facilitate similarity calculations between graphs, it is essential to identify the node composition and connections within the graphs. In this thesis, the focus is specifically on super tool level and their associated input and output data types. Consequently, a new label of a combined node is created by merging the super tools with the corresponding input and output data types in both the expert and generated workflows.

For example, in the $wfwaste\_odour$ expert workflow, a tool called *KernelDensity*(https://pro.arcgis.com/en/pro-app/latest/tool-reference/spatial-analyst/kernel-density.htm) has an input data and an output data. The Kernel Density tool adeptly computes the density of features within their surrounding neighborhood, applicable for both point and line features. After integrating the tool name, the data type of the input data and the data type of the output data, the new label of the resulting combined node is:

$'KernelDensity - ERA\_ObjectQ\_PointA + FieldQ\_RasterA\_RatioA'$

where the minus sign is followed by the CCD type of the input data and the plus sign is followed by the CCD type of the output data.

When determining the connections between these new nodes, the following logic is applied: if one of the input data in a node corresponds to the output data of another node, the two nodes are connected. It is important to note that a node can have multiple input data, and connections are established based on the relationships between the input and output data.

4.2 Complete CCD Type Annotations

In order to ensure completeness in the CCD type annotations of the expert and generated workflows, it is necessary to include all three dimensions in each data type. Produced using the APE, the generated workflows necessitate the specification of a node for each dimension, thereby ensuring that a type is assigned for each of the three semantic dimensions. However, the data types for expert workflows are not complete.Additionally, the order of the three dimensions should consistently follow the sequence: ccd:CoreConceptQ, ccd:LayerA, and ccd:NominalA.

To ensure this, a Python script is utilized, which leverages the rdflib library to manipulate the types of nodes in the graph data structure. A dictionary is generated by this script, mapping each dimension of the CCD ontology to a set of URIs that are subclasses of that dimension. This approach ensures that all three dimensions associated with each data type in the workflow are correctly identified and included in the analysis, which is crucial for tasks such as workflow comparison.

In the example in section 4.1, the reordering of the labels according to the above principle is changed to:

$'KernelDensity - ObjectQ\_PointA\_ERA + FieldQ\_RasterA\_RatioA'$

where the data type of the input data is correctly adjusted.

5 Methods

The two methods chosen are graph2vec method and graph serialization method.

5.1 Graph2vec Method

In the graph2vec method, the perspective of considering an entire graph as a document and the subgraphs within the graph as words is adopted. In this approach, the order of the words(subgraphs) and the unique labels associated with these words (subgraphs) are determined as the first step.

5.1.1 Breadth First Search Traversal Graph

In the doc2vec method, the context of a word determines its targeted word prediction. Similarly, in the graph2vec model, the ordering of subgraph labels in a graph can be viewed as the ordering of words in a document, and different ordering structures can impact the model's learning. Subgraphs are essential as they form the basis for similarity assessment, akin to matching words in document similarity. These subgraphs are identified through graph traversal, a method that systematically visits all nodes in the graph. To optimize the consistency of subgraph order between expert and generated workflows, the labels within a workflow are reordered in alignment with the workflow's steps.

In the case of the study, a workflow is specifically represented as a rooted Directed Acyclic Graph (DAG), which inherently provides a unique starting point for search. The starting point is a graph consisting of node labels synthesized using tool and CCD labels. With a defined starting point, this reordering is achieved using the breadth-first search algorithm, a widely recognized algorithm designed for traversing or exploring a graph in breadth-first order.

In this approach, the traversal starts from a given source vertex and systematically visits all the vertices at the same level in the graph before moving to the next level. The combined node containing the final output of the workflow is selected as the starting point for the traversal, considering that each workflow may have multiple inputs but always has a single final output.

5.1.2 Get Unique Labels

Uniqueness is a crucial characteristic, as it enables nodes to be compared across different workflows. The structure of the system ensures that nodes can reappear in other workflows, fostering meaningful comparisons.

In this context, the node label is constructed from the tool name coupled with one or two input data types and one output data type. When a label encompasses two input data types, the sorting order of these input data types should not influence the uniqueness of the label. In other words, labels such as "input1_data_type_input2_data_type" and "input2_data_type_input1_data_type" are considered identical at the label level. To ensure unique labels, the input data and output data within each node label are arranged separately in alphabetical order. For example,

$'Clip, ObjectQ\_PlainVectorRegionA\_CountA\_ObjectQ\_PlainVectorRegionA\_NominalA, ObjectQ\_PlainVectorRegionA\_CountA'$

After reordering,

$'Clip, CountA\_NominalA\_ObjectQ\_ObjectQ\_PlainVectorRegionA\_PlainVectorRegionA, CountA\_ObjectQ\_PlainVectorRegionA'$

5.1.3 Different Variants

To evaluate workflow similarity performance under varying conditions, 14 different node label types have been selected as variables. These variants serve a significant purpose as they enable rigorous testing of different variants of semantic workflow descriptions in the context of retrieval and similarity assessment.

Additionally, it is expected that a variant utilizing only the conceptual CCD dimension would possess the ability to generalize across syntax and toolnames, making it potentially more suitable for retrieval in QA where the question does not contain specific toolnames or syntax. This is possible because core concepts, with their associated semantic constraints, could potentially enable the synthesis of high-quality workflows[3].

Due to the rearrangement of the data type collections in section 4.2, the combination of tool names and data type collections can be easily obtained:

1. toolname and coreconcept
2. toolname and layer type
3. toolname and measurement level
4. toolname, coreconcept and layer type
5. toolname, coreconcept and measurement level
6. toolname, layer type and measurement level
7. toolname, coreconcept, layer type and measurement level
8. coreconcept
9. layer type
10. measurement level
11. coreconcept and layer type
12. coreconcept and measurement level
13. layer type and measurement level
14. coreconcept, layer type and measurement level

5.1.4 Doc2vec Model

The Doc2Vec algorithm, available in the gensim library, is utilized for generating vector representations of entire documents, enabling the acquisition of document-level embeddings.

Prior to computing the similarity between the embeddings of the expert workflows and the generated workflows, it's essential to fine-tune the parameter values within the Doc2Vec function. This fine-tuning aims to yield the optimal similarity. The parameters of the Doc2Vec function significantly influence the model's overall performance and the caliber of the resulting vector

representations. Here are the optimal parameters that have been determined through the training process:

1. ‘document_collections‘: This is the list of graphs to train on. Each graph is represented as a list of labels of nodes. In this case, all the expert workflows and generated workflows are involved in this training set.

2. ‘vector_size=128‘: This parameter defines the size of the output vectors (embeddings) learning for each graphs. In this case, the embeddings will be 128-dimensional.

3. ‘dm=0‘: When dm=0, ’distributed bag of words’ (PV-DBOW) which is an instance of the skipgram model is used.

4. ‘dbow_words=1‘: If set to 1 (default 0), while training doc2vec’s DBOW (‘dm=0‘) concurrently trains word(subgraph) vectors – effectively behaving as skip-gram.

5. ‘window=5‘: The ‘window‘ parameter defines the context window size. The window size here is 5, it means 5 words before the current word and 5 words after are taken into account while training.

6. ‘min_count=1‘: This ignores all words with total frequency lower than 1, which means all low-frequency words are considered.

7. ‘sample=0.1‘: This threshold determines the configuration for randomly downsampling higher-frequency words. A smaller value increases the likelihood of downsampling each high-frequency word.

8. ‘workers=4‘: This is the number of worker threads to use for training.

9. ‘epochs=10‘: Number of iterations (epochs) over the corpus.

10. ‘alpha=0.5‘: The initial learning rate.

5.1.5 Cosine Similarity

After the graph embeddings for each workflow have been obtained, the cosine similarity between the embeddings of the expert and generated workflows should be computed.

Cosine similarity is a measure used to gauge the degree of similarity between two vectors, within the confines of a multi-dimensional space. This metric is determined by calculating the cosine of the angle between these two vectors, hence its name 'cosine similarity'. In the context of our discussion on workflow embeddings, it can be employed to assess the level of similarity between an expert workflow and a generated workflow. The values of cosine similarity lie between -1 and 1, with a value close to 1 indicating a high degree of similarity and a value near -1 suggesting a high degree of dissimilarity between the two workflows.

5.2 Graph Serialization Method

During the serialization process of a graph, the labels assigned to the nodes are converted into strings and concatenated in a specific order to create a serialized representation. This order is determined using the breadth-first search method, which is the same as described in part 5.1.1. Additionally, the node labels are internally reordered using the alphabetical reordering method, as explained in part 5.1.2, to obtain new unique node labels. Finally, the new labels of each node are concatenated into a string following the breadth-first search order, resulting in a serialized representation of the graph.

After serializing all the expert workflows and the generated workflows, the next step is to employ the edit distance method to measure the similarity between the two workflows. In this case, the Levenshtein distance method is chosen to calculate the edit distance between the two serialized representations of the graph. The Levenshtein distance method is a way to measure the similarity between two strings. It calculates the minimum number of operations required to transform one string into another.A high Levenshtein Distance means that a significant number of edits are needed to make the two strings identical, indicating a high level of dissimilarity. Conversely, a

lower Levenshtein Distance is preferable.

To obtain results on workflow similarity under different variants, specific variables were selected, as described in Section 5.1.3.

5.3 Analysis and Evaluation Approach

5.3.1 Analysis Approach for the Graph2Vec Method

The cosine similarity between expert and generated workflow embedding are analyzed from four distinct measurements for varying node label variables:

- For each expert workflow, find the generated workflow with the largest similarity to it and get the average of all these largest similarities.
- For each expert workflow, find the generated workflow with the smallest similarity to it and get the average of all these smallest similarities.
- For each expert workflow, get its median similarity and get the average of all these median similarities.
- For each expert workflow, get its mean similarity and get the average of all these mean similarities.

Analyzing cosine similarity from these four perspectives provides a comprehensive understanding of different variants of semantic workflow descriptions in terms of similarity assessment. Evaluating the largest and smallest similarities highlights the potential and limitations of the generated workflows. The median similarity provides a robust measure of typical performance, being less sensitive to outliers, while the mean similarity provides an average performance measure across all workflows.

5.3.2 Analysis Approach for the Graph Serialization Method

The levenshtein distance between the serialized strings of expert workflows and generated workflows is analyzed from four distinct measurements for varying node label variables, similar to the previous analysis:

- For each expert workflow, find the generated workflow with the shortest distance to it and get the average of all these shortest distance.

- For each expert workflow, find the generated workflow with the longest distance to it and get the average of all these longest distance.
- For each expert workflow, get its median distance and get the average of all these median distances.
- For each expert workflow, get its mean distance and get the average of all these mean distances.

When analyzing the Levenshtein Distance between two strings, not only the distance value but also the length of the strings are considered to determine the level of similarity or dissimilarity between the strings. As an illustration of this, the variant containing both tool names and full data types will be selected to demonstrate the effectiveness of the Levenshtein Distance method in evaluating workflow similarity.

5.3.3 Evaluation Approach

The effectiveness of the similarity measure is primarily demonstrated through the evaluation process. Given that experts have arrived at judgments in assessing workflow similarity, this process provides a reliable benchmark for the assessment.

The evaluation metrics chosen here are based on the manual comparison of workflows to assess retrieval quality. In the variant to be derived from the graph2vec method, it is planned to select three generated workflows with the highest, median, and lowest similarity to each expert workflow. The QuAnGIS project team, including GIS experts, manually selected the generated workflow with the highest similarity to each expert workflow to validate the retrieval effect. The order of these three workflows was randomized to enable unbiased assessments.

6 Example Description for Graph2vec Method

Let's use the "wfwaste_odour" expert workflow, depicted in Figure 2, as an illustrative example to elucidate the entire analysis process.

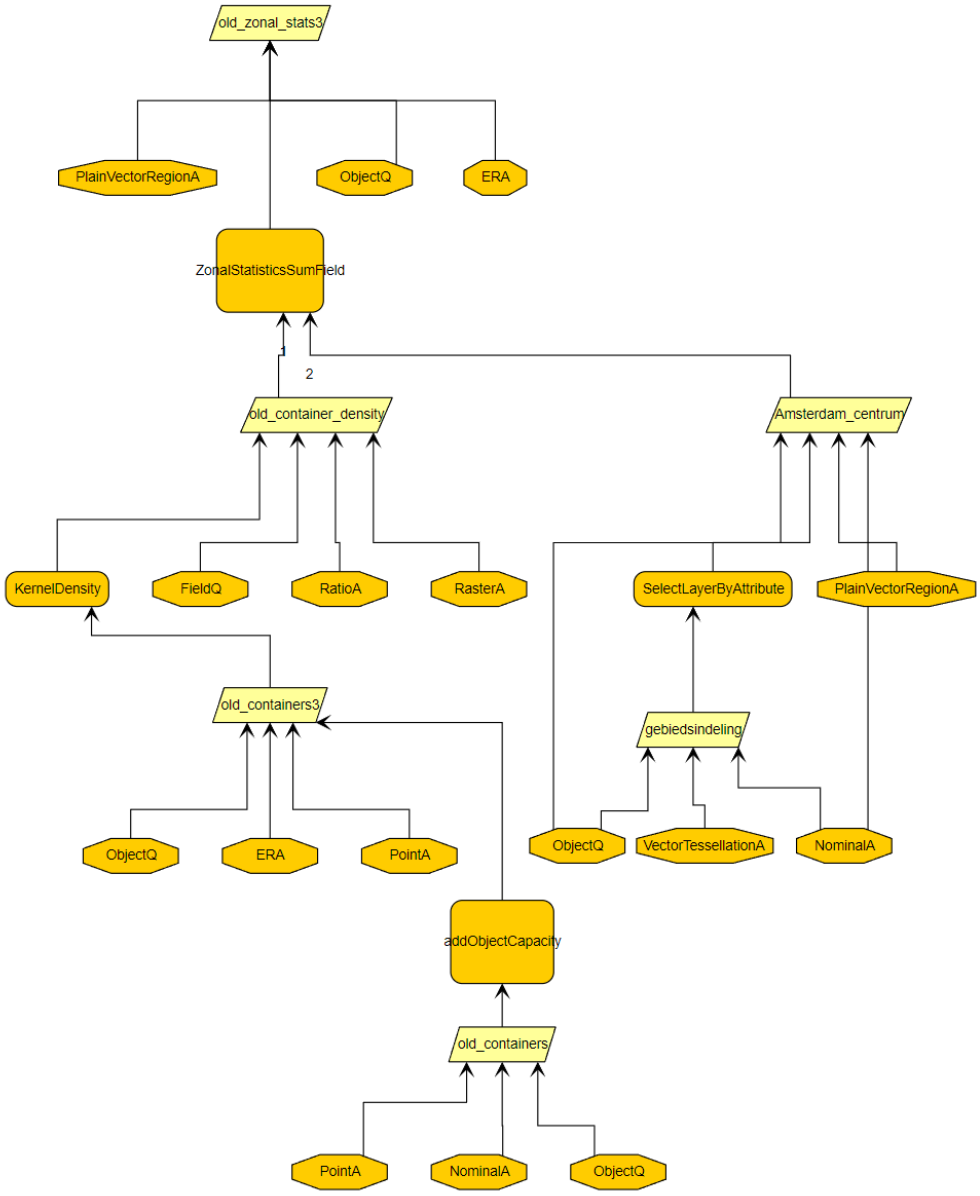

Figure 2: wfwaste_odour workflow

1. Composite nodes are extracted, as shown in Figure 3, which include the tool name, the CCD types of both input and output data, as well as the connections between these nodes. By employing the methodol-

ogy articulated in section 4.2, the labels of these four composite nodes are complete and arranged according to their respective data types. A preceding minus sign denotes the input data, while a following plus sign indicates the output data. These composite nodes are as follows:

$1 :' SelectLayerByAttribute-ObjectQ\_VectorTessellationA\_NominalA+ ObjectQ\_PlainVectorRegionA\_NominalA'$,
$2 :' ZonalStatisticsSumField - FieldQ\_RasterA_{R}atioA - ObjectQ\_ PlainVectorRegionA\_NominalA'$,
$3 :' addObjectCapacity-ObjectQ\_PointA\_NominalA+ObjectQ\_PointA_{E}RA'$,
$4 :' KernelDensity-ObjectQ\_PointA\_ERA+FieldQ\_RasterA\_RatioA'$

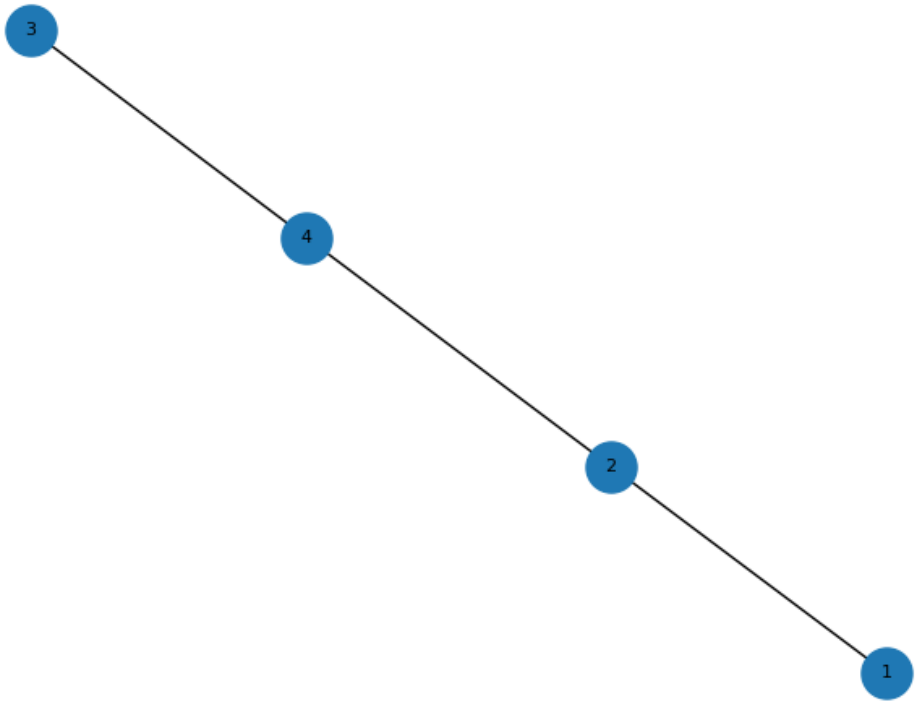

Figure 3: Combined nodes and connections

2. The above shows that the labels of the current nodes are not sorted according to the steps within the workflow. So to maximize the align-

ment in the label order between the expert workflow and the generated workflow, the traversal method described in Section 5.1.1 is utilized. From the initial $wfwaste\_odour$ graph, it's apparent that the 'ZonalStatisticsSumField' tool incorporates the final output old_zonal_stats3, hence this composite node is selected as the traversal's starting point. After reordering the labels of this workflow, we obtain a new sequence:

$1 :' ZonalStatisticsSumField - FieldQ\_RasterA\_RatioA - ObjectQ\_PlainVectorRegionA\_NominalA'$,
$2 :' SelectLayerByAttribute{-}ObjectQ\_VectorTessellationA\_NominalA + ObjectQ\_PlainVectorRegionA\_NominalA'$,
$3 :' KernelDensity{-}ObjectQ\_PointA\_ERA + FieldQ\_RasterA\_RatioA'$,
$4 :' addObjectCapacity{-}ObjectQ\_PointA\_NominalA + ObjectQ\_PointA\_ERA'$

It should be noted that the 'ZonalStatisticsSumField' tool has two neighboring nodes, 'SelectLayerByAttribute' and 'KernelDensity'. Consequently, during the breadth-first traversal, the order of these two nodes may be interchanged, that is, 'ZonalStatisticsSumField', 'KernelDensity', 'SelectLayerByAttribute', 'addObjectCapacity'. This minor inconsistency will be addressed further in the reflection section.

3. Get the unique label according to the method described in section 5.1.2.

$1 :' ZonalStatisticsSumField{-}FieldQ\_NominalA\_ObjectQ\_PlainVectorRegionA\_RasterA\_RatioA'$,
$2 :' SelectLayerByAttribute{-}NominalA\_ObjectQ\_VectorTessellationA + NominalA\_ObjectQ\_PlainVectorRegionA'$,
$3 :' KernelDensity{-}ERA\_ObjectQ\_PointA + FieldQ\_RasterA\_RatioA'$,
$4 :' addObjectCapacity{-}NominalA\_ObjectQ\_PointA + ERA\_ObjectQ\_PointA'$

4. The aforementioned three steps are applied to all expert workflows and the generated workflows. This procedure yields a sequence of labels for each workflow, which is subsequently input into the doc2vec model. Once the doc2vec model generates the graph embeddings for each workflow, the cosine similarity between these embeddings is computed.

5. The same steps are applied for the remaining six variables.

7 Results

7.1 Graph2vec Method for Comparing GIS Workflow Similarity

The similarity results are analyzed by comparing and averaging the similarities between the expert workflows and the generated workflows, providing an assessment of the degree to which the generated workflows align with the expert workflows.

Fourteen tables, each corresponding to one of the fourteen variants, are obtained and can be accessed at the following website:[11]. Table 1, specifically, displays the results for the variant that exhibits the highest average maximum similarity across all variants.

Table 1: Variable: Coreconcept and layer types of input data and output data without tool name

Coreconcept and Layer Types	
Measurements	Similarity Value
Average Largest Similarity	0.77
Average Smallest Similarity	-0.13
Average Median Similarity	0.35
Average Mean Similarity	0.34

From the results presented in Tables 1, the following inference is drawn:

- The variant which includes the core concept and layer type in the CCD type set, without incorporating the tool name, exhibits the highest average maximum similarity among all, reaching up to 0.77. Although its average minimum similarity may seem slightly poorer compared to other variants, the median similarity and mean similarity still attest to its generally good performance. This variant primarily demonstrates the method's ability to bridge workflows that have different tool names but similar transformation semantics.

7.2 Graph Serialization Method for Comparing GIS Workflow Similarity

The serialization method for graphs serves as a comparative measure to the graph embedding method. The results for the 14 corresponding Levenshtein Distance, each based on one of the 14 variants, can be accessed at the following website:[11]. As an example to elaborate on the results, the variant that incorporates both the tool name and the complete data type set is presented, as shown in Table 2.

Table 2: Variable: Tool name with full types including core concept, layer type, measurement level of input data and output data

Tool Name with Full Types	
Measurements	Levenshtein Distance
Average Shortest Distance	232.36
Average Longest Distance	397.21
Average Median Distance	292.71
Average Mean Distance	295.39

From the results presented in Tables 2, the following inference is drawn:

- The serialized strings for both expert workflows and generated workflows primarily have lengths ranging from 300 to 500. However, even within this range, the average shortest Levenshtein Distance recorded is 232.36. This suggests that, on average, over 232 edits are needed to change one string into the other. This represents a substantial proportion of the total length of the strings, indicating a high degree of dissimilarity between the expert and generated workflows.

- From this observation, it is evident that the graph serialization method, when applied to serialized strings, does not assess the similarity between

expert and generated workflows as effectively as the graph embedding method does.

8 Evaluation for Graph2vec Method

Through the comparison of the similarity of expert and generated workflows using the graph2vec method, it appears that the variant that incorporates core concepts and layer types, but excludes tool names, generalizes best over workflows with similar concepts but different tools. Therefore, this variant is utilized in the evaluation for the retrieval and similarity assessment in a QA system. The results of this manual evaluation are presented in Table 3. The source data for evaluation and the final evaluation document an be found at the following website:[11]

Table 3: Results of manually selecting the workflow with the highest similarity

Variant: Core concept and Layer Types		
Expert Workflow	Most Similar Generated Workflow	Manual Selection of The Most Similar Generated Workflow
Expert1	solution148	solution148
Expert2	solution49	solution49
Expert3	solution86	solution86
Expert4	solution38	solution38
Expert5	solution151	solution104
Expert6	solution151	solution151
Expert7	solution35	solution35
Expert8	solution142	solution142
Expert9	solution26	solution26
Expert10	solution40	solution141
Expert11	solution49	solution49
Expert12	solution63	solution164
Expert13	solution148	solution148
Expert14	solution97	solution97

From the results obtained in Table 3, the following inferences can be drawn:

- Compared to the manually selected most similar generated workflows, the accuracy of the best workflows obtained through the graph2vec method reached 85.7%. This highlights the effectiveness of this approach in workflow retrieval within a QA system.
- Concurrently, in terms of variant selection, the variant that includes core concepts and layer types but excludes tool names primarily demonstrates the method's ability to bridge workflows that have different tool names but similar transformation semantics.

9 Conclusions

In this study, the graph2vec model from the graph embedding methods is employed to learn workflow embeddings, with cosine similarity utilized to evaluate the similarity between expert and generated workflows. Concurrently, the effect of different variants of semantic workflow descriptions on retrieval and similarity assessment in a QA system is explored. By manually comparing workflows to evaluate retrieval quality, the efficacy of the graph2vec model in comparing workflows is demonstrated.

In addition, another graph serialization method is chosen for comparison, to investigate the effects of different methods on the retrieval of GIS workflows. Although both methods consider the internal structure of the graph, the graph serialization method falls short in the retrieval and similarity assessment of workflows in the QA system when compared to the graph embedding method.

All codes and data used in this study can be found at the following website:[11]

10 Discussion

This study has made notable progress in the application of vector embedding methods, specifically the graph2vec model, for assessing workflow similarity. However, it is important to acknowledge that the current method still has room for improvement.

One of the key observations from the evaluation of the results is that even the most similar workflows are often not adequate for substitution due to notable differences. The current pool of generated workflows does not always contain equivalent workflows for the expert tasks, which suggests that the validity of similarity assessments is therefore limited. However, this does not undermine the validity of the similarity assessment method itself.

A further area for potential improvement is the integration of this method into a real Question-Answering (QA) or retrieval system. The current study has demonstrated the potential of the graph2vec model for assessing workflow similarity, but the practical application of this method in a real-world system is yet to be explored. This could involve integrating the graph2vec model into an existing QA system to enhance its ability to handle tasks related to workflow similarity.

In conclusion, while the current method has shown promise, further research and development are needed to fully realize its potential in the realm of workflow similarity assessment and retrieval.

References

[1] A. Omran, S. Dietrich, A. Abouelmagd, and M. Michael, "New arcgis tools developed for stream network extraction and basin delineations using python and java script," *Computers & Geosciences*, vol. 94, pp. 140–149, 2016.

[2] A. A. Shah, S. D. Ravana, S. Hamid, and M. A. Ismail, "Accuracy evaluation of methods and techniques in web-based question answering systems: a survey," *Knowledge and Information Systems*, vol. 58, pp. 611–650, 2019.

[3] S. Scheider, E. Nyamsuren, H. Kruiger, and H. Xu, "Geo-analytical question-answering with gis," *International Journal of Digital Earth*, vol. 14, no. 1, pp. 1–14, 2021.

[4] J. F. Kruiger, V. Kasalica, R. Meerlo, A.-L. Lamprecht, E. Nyamsuren, and S. Scheider, "Loose programming of gis workflows with geo-analytical concepts," *Transactions in GIS*, vol. 25, no. 1, pp. 424–449, 2021.

[5] S. Scheider, R. Meerlo, V. Kasalica, and A.-L. Lamprecht, "Ontology of core concept data types for answering geo-analytical questions," *Journal of Spatial Information Science*, no. 20, pp. 167–201, 2020.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[7] R. Bergmann and Y. Gil, "Similarity assessment and efficient retrieval of semantic workflows," *Information Systems*, vol. 40, pp. 115–127, 2014.

[8] S. Beco, B. Cantalupo, L. Giammarino, N. Matskanis, and M. Surridge, "Owl-ws: a workflow ontology for dynamic grid service composition," in *First International Conference on e-Science and Grid Computing (e-Science'05)*. IEEE, 2005, pp. 8–pp.

[9] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "graph2vec: Learning distributed representations of graphs," *arXiv preprint arXiv:1707.05005*, 2017.

[10] L. Wen, T. Amagasa, and H. Kitagawa, "An approach for xml similarity join using tree serialization," in *Database Systems for Advanced Applications: 13th International Conference, DASFAA 2008, New Delhi, India, March 19-21, 2008. Proceedings 13*. Springer, 2008, pp. 562–570.

[11] Github address. [Online]. Available: https://github.com/Sarasara00/thesis_anqi.git

[12] V. Kasalica and A.-L. Lamprecht, "Workflow discovery with semantic constraints: The sat-based implementation of ape," *Electronic Communications of the EASST*, vol. 78, 2020.